

**Cognizant Academy**

**Metalapps Sales Automation**

**C#, ADO .Net Knock Out Challenge**

**Version 2.0**

	<b>Prepared By / Last Updated By</b>	<b>Reviewed By</b>	<b>Approved By</b>
<b>Name</b>	Sajitha Nazer	Senthil/Velu	
<b>Role</b>			
<b>Signature</b>			
<b>Date</b>			

## Table of Contents

<b>1.0</b>	<b>Introduction</b>	<b>1</b>
1.1	Purpose of this document	1
1.2	Definitions & Acronyms	2
1.3	Project Overview	2
1.4	Scope	2
1.5	Target Audience	2
1.6	Software Requirement	3
1.6.1	Software Requirements	3
<b>2.0</b>	<b>Functional Requirements</b>	<b>3</b>
2.1	Functional Requirements	3
2.2	Use case Diagram	4
2.3	System Architecture Diagram	5
<b>3.0</b>	<b>Design Specification</b>	<b>6</b>
3.1	Data Design	6
3.2	Component Details for identified Use Cases	7
3.2.1	Get sales data and calculate the net sales amount for each sales	7
3.2.2	Add sales details to database	8
3.3	Component Specification	8
3.3.1	AddSalesDetails Method	9
3.3.2	CalculateNetAmount Method	10
3.3.3	GetConnection Method	10
3.4	General Design Constraints	10
<b>4.0</b>	<b>Submission</b>	<b>11</b>
4.1	Code submission instructions	11
<b>5.0</b>	<b>Change Log</b>	<b>11</b>
<b>6.0</b>	<b>Evaluation Areas</b>	<b>11</b>

## 1.0 Introduction

### 1.1 Purpose of this document

Metalapps is a distributor for the newly arrived Steam Ovens that offer a healthy method of cooking. While steam ovens have recently gained popularity in residential kitchens, there has been an increase in sales this year due to High-volume purchase orders from wholesalers. As a result Metalapps was facing a huge work load on the billing process of their sales to wholesalers.

Storing the sales details such as the customer name, number of units sold, calculating the discount amount and net payable amount, storing the sales details and more process were done manually. So, Metalapps decides to outsource the billing process of their sales to Global Tek Software Company. Help the Global Tek to automate the above task.

The wholesale rate of 1 unit of Steam Oven is **75,350**

The minimum number of purchase should be more than 5 and there is no maximum limit.

Based on the number of products purchased a discount is offered from the total amount.

Metalapps has the following business activities that must be automated.

1. Get the sales details from the customer
2. Calculate the net amount based on the number of units sold and discount specified.
3. Store the sales details for each sale.

## 1.2 Definitions & Acronyms

Definition / Acronym	Description
Req	Requirement

## 1.3 Project Overview

This project captures the various concepts, techniques and skills learnt and helps to put them into practice using C# with ADO.NET. Admittedly, this would be at a scaled-down level since the purpose is to let the associate experience the various concepts learned in C# as an individual. The individual associate is expected to create a console-based application within the specified time.

## 1.4 Scope

The scope of the system is explained through its following modules

1. Get the sales details, calculate the total sales amount based on the number of units purchased and the applicable discount for each sale.
2. Store the sales details of each sale.

## 1.5 Target Audience

Learner Level

## 1.6 Software Requirement

### 1.6.1 Software Requirements

#	Item	Specification/Version
1.	C#	6
2.	ADO.NET	4.5
3	SQLSERVER	2017

**Note:** All the required hardware and software will be provided in the Tekstac platform.

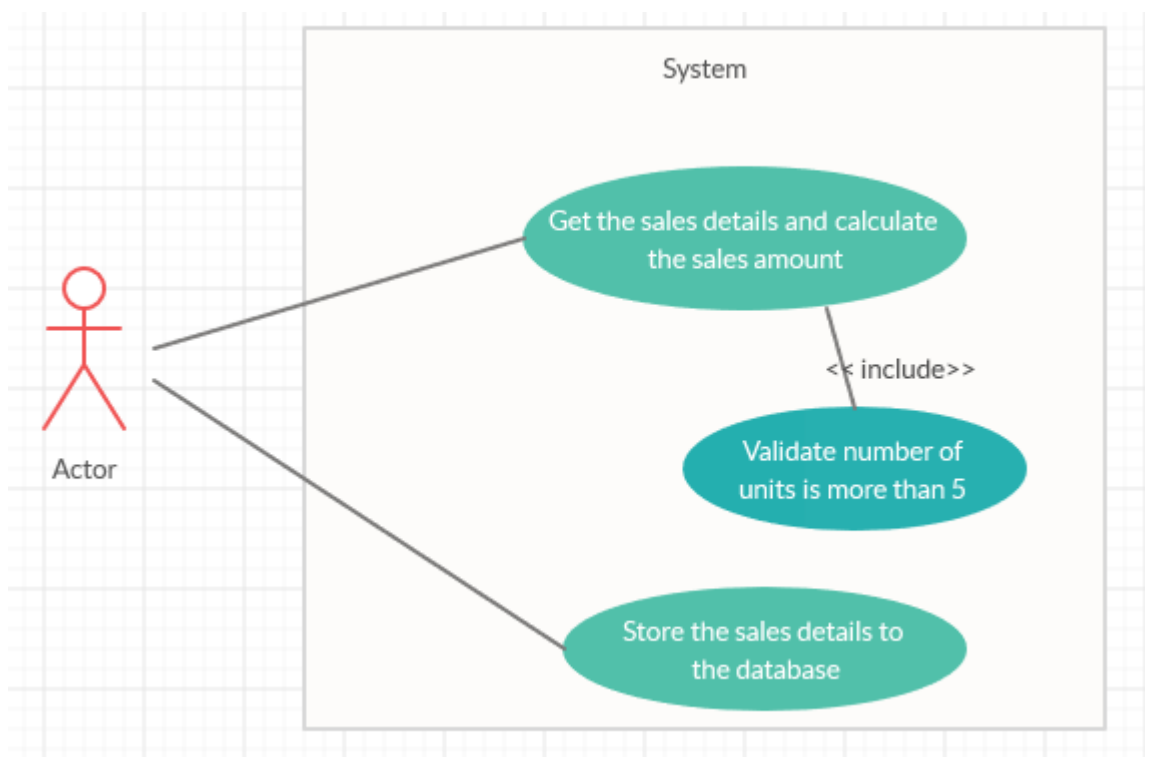
## 2.0 Functional Requirements

### 2.1 Functional Requirements

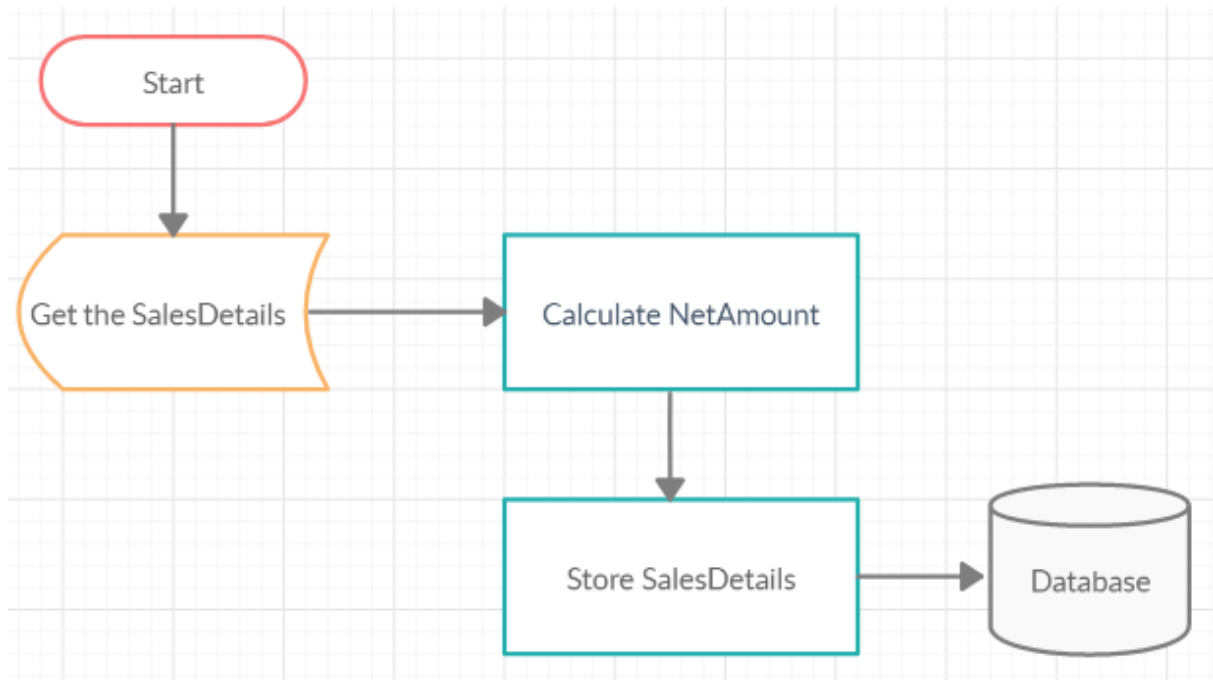
<b>Req. #</b>	1
<b>Req. Name</b>	Get the sales details and calculate the net amount.
<b>Req. Description</b>	The number of units purchased by the customer and the other details of the sales are entered by the Manager. Calculate the net sales amount based on the number of units sold and the discount percentage.
<b>Actors/ Users</b>	Manager
<b>Comments</b>	The Metalapps company sales manager is responsible for getting the sales data and calculating the net amount.

<b>Req. #</b>	2
<b>Req. Name</b>	Store the SalesDetails for each sales to the database
<b>Req. Description</b>	After calculating the net amount, SalesDetails is stored into the database.
<b>Actors/ Users</b>	Manager
<b>Comments</b>	The Metalapps company sales manager is responsible for adding the sales details for each sales into the database

## 2.2 Use case Diagram



## 2.3 System Architecture Diagram



## 2.4 Sample Input/Output

### SAMPLE INPUT :

```

Enter sales id
1111
Enter customer name
Remo Traders
Enter the number of units sold
10
  
```

### SAMPLE OUTPUT :

```

Sales Bill
*****
Sales Id : 1111
Customer Name : Remo Traders
Number of Units Sold : 10
Net Amount : 738430
  
```

```

Display data
from
Object itself
  
```

## 3.0 Design Specification

### 3.1 Data Design

#### Table Structure:

Table name: SalesDetails	
Column Name	Data type
Sales_id	int
Customer_name	varchar
Noof_units	int
Net_amount	double

#### Design Constraints:

- Use **SQLSERVER** database to store the data. The database name is "SalesDatabase". **This is already created for you in Tekstac.**
- The table names and the column names should be the same as specified in the table structure.
- The database connection information is specified in the "App.config" file, which is also provided as part of code skeleton. THIS IS GIVEN ONLY FOR YOUR REFERENCE. You need NOT change this.
- *Use the connection string name "SqlCon" for creating database connection(refer App.config).*

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
  </startup>
  <connectionStrings>
    <add name=" SqlCon"
    connectionString="server=localhost;database=SalesDatabase;uid=XXXXX;password=XXXXXX;"/>
  </connectionStrings>
</configuration>
```

**Note:** The code skeleton will be available in the Tekstac platform



## 3.2 Component Details for identified Use Cases

### 3.2.1 Get sales data and calculate the net sales amount for each sales

In the 'Main' get the following values,

SalesId
CustomerName
NoOfUnits

The wholesale rate of 1 unit Steam Oven is **75,350**

Calculate the discount amount based on the number of units purchased by the customer.

Generate the sales bill for each sales based on the units purchased and discount amount.

**Hint:**

NetAmount=0( initial amount , later calculated and assigned)

Create a SalesDetails object and assign the values to the properties. Calculate the net amount and assign the NetAmount based on the conditions given below.

Based on the number of products purchased a discount is offered from the total amount.

For sample input/output, refer section 2.4.

Calculate the sales based on the below condition:

Number of products purchased	Percentage of discount from the total sales amount
<=5	0
>5 and <=10	2%
>10 and <= 15	5%
>15 and <=20	8%
>20 and above	10%

**For example:** If the units purchased are **8** then the discount is 2% from the total sales amount.

The net amount is calculated as Rs. 590744

That is,

$$((75,350 * 8) - (75,350 * 8) * 2\%)$$

#### Validation 1:

The minimum number of units purchased should be more than 5. If a number less than or equal to 5 is provided for NoOfUnits, it should throw '**ArgumentOutOfRangeException**' with a message "**No Sales for units below 5**".

### 3.2.2 Add sales details to database

Add the sales details to the database.

## 3.3 Component Specification

**Class Name : SalesDetails (model class)**

#### Responsibility:

This model object holds the state of the sales detail at all point-in-time.

Type(Class)	Fields	Properties
SalesDetails	<b>int</b> salesId; <b>string</b> customerName; <b>int</b> noOfUnits; <b>double</b> netAmount;	<b>int</b> SalesId; <b>string</b> CustomerName; <b>int</b> NoOfUnits; <b>double</b> NetAmount;

**Note:** Keep all the fields 'private' and properties 'public'.

### Class Name : Metalapps (utility class)

#### Responsibility:

This class has the supporting methods to automate the sales process given in our scope. This class includes the methods to calculate sales net amount and add the details to database.

Type(Class)	Field	Property
Metalapps	Nil	<b>public SqlConnection</b> SqlCon;

**Note:** This 'SqlCon' must be assigned with the object value when the connection is created for the first time.

### Class Name : Metalapps (utility class)

#### 3.3.1 AddSalesDetails Method

Type(Class)	Method	Responsibilities
Metalapps	void AddSalesDetails(SalesDetails sd)	This method should accept an SalesDetails object and <b>execute an sql query</b> to insert the details into the database

### 3.3.2 CalculateNetAmount Method

Type(Class)	Method	Responsibilities
Metalapps	void CalculateNetAmount(SalesDetails details)	This method accepts a SalesDetails object with assigned values (The initial value for NetAmount alone should be kept zero) as an argument. Calculate the discount amount based on the number of units sold and the conditions specified. The calculated NetAmount is assigned back to the input object

**Class Name : DBHandler(DAO class)**

### 3.3.3 GetConnection Method

**Responsibility:**

This method should connect to the database by reading the database details from the *App.config* file and it should return the connection object.

Type(Class)	Method	Resources
DBHandler	SqlConnection GetConnection()	App.config file contains the database connection details.

## 3.4 General Design Constraints

1. The fields/properties/method/class name should be correctly specified as given in the document.
2. Keep all the classes as '**public**'

3. Do not change the App.config file.
4. Do not change the namespace name.

## 4.0 Submission

### 4.1 Code submission instructions

1. Do not change the code skeleton given as your code will be auto evaluated.
2. You can validate your solution against sample test cases during the assessment duration.
3. Your last submitted solution will be considered for detailed evaluation.
4. Make sure to submit the solution before the time limit. After the assessment duration you will not be allowed to submit the solution.

## 5.0 Change Log

	Changes Made			
V1.0.0	Initial baseline created on <dd-Mon-yy> by <Name of Author>			
Vx.y.z	<Please refer the configuration control tool / change item status form if the details of changes are maintained separately. If not, the template given below needs to be followed>			
	Secti on No.	Chang ed By	Effecti ve Date	Changes Effected

## 6.0 Evaluation Areas

S.No	Description
1.	Declaration of attributes, properties and methods in the class

	SalesDetails
2.	Declaration of properties and methods in the class DBHandler
3.	Implementation to create a valid database connection object
4.	Implementation to add sales details to database
5.	Calculation of net amount for units purchased based on various discounts.
6.	Validate number of units is more than 5.